

**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

Claims 1 to 21 (cancelled).

Claim 22 (currently amended): A control system for a device, wherein a graphic user interface (“GUI”) controller operates a GUI of the device independently from an embedded controller of the device, the control system comprising:

the embedded controller for controlling and monitoring the device;

a liquid crystal display (“LCD”) for displaying the GUI to a user, the GUI including:

a first GUI object ~~for displaying~~ comprising a graphical presentation of a status of the device; and

a second GUI object ~~for displaying~~ comprising a graphical presentation of a command to the device;

a touch screen for detecting the command from the user;

the GUI controller, comprising:

at least one memory, comprising:

a document buffer storing a document defining an appearance of the GUI, the document comprising:

a first operation code (“opcode”) identifying the first GUI object;

a second opcode identifying [[the]] ~~a~~ source of the status;

a third opcode identifying the second GUI object; and

a fourth opcode identifying [[the]] a destination of the command;

a data buffer storing the status and the command;

a GUI object library storing:

a first set of executable codes defining an appearance and a functionality of the first GUI object, the first set of executable codes comprising with instructions for rendering the first GUI object, receiving non-graphical data of the status from the embedded processor controller, and further rendering the first GUI object to show a visual response to a change to the status;

a second set of executable codes defining an appearance and a functionality of the second GUI object, the second set of executable codes comprising with instructions for rendering the second GUI object, receiving the command from the touch screen, further rendering the second GUI object to show a visual response to a change to the command, and sending non-graphical data of the command to the embedded processor controller;

a frame buffer storing at least one complete display frame image of the GUI;

a GUI processor for rendering the GUI and handling user inputs independently from the embedded controller, wherein:

the GUI processor is coupled to the touch screen via a touch screen analog to digital converter to receive the command from the user, wherein the touch screen is not directly connected to the embedded controller;

the GUI processor is coupled to the embedded processor controller via a serial UART interface to send the non-graphical data of the

command to the embedded controller and to receive the non-graphical data of the status from the embedded controller;

the GUI processor is coupled to the at least one memory via a ~~direct~~ memory bus interface, wherein:

in response to the first and the second opcodes, the GUI processor executes the first set of executable codes to render the first GUI object to the frame buffer independently from the embedded ~~processor controller~~, ~~to communicate with the embedded processor~~ to receive the non-graphical data of the status from the embedded processor controller, and to further render the first GUI object to the frame buffer to show a visual response in response to a change to the status independently from the embedded processor controller;

in response to the third and the fourth opcodes, the GUI processor executes the second set of executable codes to render the second GUI object to the frame buffer independently from the embedded ~~processor controller~~, to receive the command from the touch screen independently from the embedded ~~processor controller~~, to further render the second GUI object again to the frame buffer to show a visual response to a change to the command independently from the embedded processor controller, and to send the non-graphical data of the command to the embedded processor controller;

a pixel serializer coupled to the LCD to continuously refresh the LCD with the complete display frame image in the frame buffer that contains both the rendered first GUI object and the rendered second GUI object.

Claim 23 (currently amended): A first controller providing a graphical user interface (“GUI”) for a device independently from a second controller that monitors and controls the device, the first controller comprising:

at least one memory comprising:

a document buffer storing a document, the document comprising ~~an a first~~ operation code (“opcode”) identifying a GUI object in the GUI and a second opcode identifying a parameter of the GUI object, the GUI object being a graphical presentation of the parameter, the parameter being from ~~a sourcee external to the first~~ the second controller;

a frame buffer for storing at least one complete frame of the GUI including ~~a visual representation of the GUI object~~;

a GUI object library storing executable codes defining an appearance and a functionality of the GUI object, the executable codes comprising with instructions for ~~communicating receiving non-graphical data of the parameter with from~~ the sourcee the second controller and rendering the GUI object in response to any change to the parameter;

a processor coupled to the second controller and the at least one memory, wherein the processor (1) reads the first and the second opcodes, (2) reads the executable codes, (3) executes the executable codes to ~~communicate receive the non-graphical data of the parameter with the sourcee from the second controller~~ and to render the GUI object in response to any change to the parameter independently from the second controller, and (4) saves the rendered GUI object in the frame buffer.

Claim 24 (previously presented): The first controller of claim 23, further comprising an output device coupled to the frame buffer to receive the GUI, the output device displaying the GUI to the user.

Claim 25 (previously presented): The first controller of claim 24, wherein the output device is a liquid crystal display (“LCD”).

Claim 26 (previously presented): The first controller of claim 25, further comprising a pixel serializer coupled between the frame buffer and the LCD, the pixel serializer outputting each line of the GUI in the frame buffer to the LCD.

Claims 27 to 30 (canceled).

Claim 31 (currently amended): The first controller of claim 23, wherein ~~the source external to the first controller is the second controller, the parameter is a status of the device from the second controller to the user, and the processor communicates the non-graphical parameter with the second controller source by receiving the status from the second controller.~~

Claim 32 (previously presented): The first controller of claim 31, wherein the GUI object is a text field.

Claim 33 (currently amended): The first controller of claim 23, further comprising another memory coupled to the processor, the another memory storing the document, the processor ~~buffering~~ loading the document from the another memory to the at least one memory.

Claim 34 (currently amended): The first controller of claim 23, wherein the second controller further comprises another memory storing the document, the second controller reading the document from the another memory and sending the document to the first controller, the first controller ~~storing~~ loading the document in the at least one memory.

Claim 35 (currently amended): A method for a first controller to generate a graphic user interface (“GUI”) for a device independently from a second controller that monitors and controls the device, comprising:

~~writing~~ loading a document [[in]] into a document buffer in at least one memory, wherein the document ~~defines~~ defining an appearance of the GUI, the document comprising a first operation code (“opcode”) identifying a GUI object and a second opcode identifying a parameter of the GUI object, the GUI object being a graphical presentation of the parameter, the parameter being from ~~a source external to the first~~ the second controller;

in response to the first and the second opcodes, retrieving executable codes of the GUI object from a GUI object library stored in the at least one memory, ~~wherein the executable codes~~ comprises ~~instructions that define~~ defining an appearance and a functionality of the GUI object with, the executable codes comprising ~~instructions for communicating~~ receiving ~~non-graphical data of the parameter with the source from the second controller~~ and rendering the GUI object in response to any change to the parameter;

independently from [[a]] the second controller, executing the instructions to communicate receive the non-graphical data of the parameter with the sourcee from the second controller and to render the GUI object in response to any change to the parameter;

writing the rendered GUI object in a frame buffer in the at least one memory; and sending the rendered GUI object from the frame buffer to an output device, wherein the output device displays the GUI to a user.

Claim 36 (previously presented): The method of claim 35, wherein the output device is a liquid crystal display (“LCD”).

Claim 37 to 40 (canceled).

Claim 41 (currently amended): The method of claim 35, wherein ~~the sourcee external to the first controller is the second controller~~, the parameter is a status of the device from the second controller to the user, ~~, and the first controller communicates the non-graphical parameter with the sourcee by receiving the status from the second controller~~.

Claim 42 (previously presented): The method of claim 41, wherein said rendering the GUI object comprises drawing a text field.

Claim 43 (currently amended): The method of claim 35, further comprising reading the document from another memory ~~and buffering prior to said loading~~ the document [[in]] into the at least one memory.

Claim 44 (currently amended): The method of claim 35, further comprising receiving the document from the second controller ~~and buffering prior to said loading~~ the document [[in]] into the at least one memory.

Claim 45 (withdrawn): A method for programming a graphic user interface (“GUI”) for a GUI controller of a device, comprising:

creating a HTML page defining an appearance of the GUI, said creating comprising:

writing an applet code for a GUI object;

writing a reference to a source of a parameter of the GUI object;

compiling the HTML page into a compact HTML page, said compiling comprising

replacing the applet code with a first operation code (“opcode”) identifying the GUI object;

replacing the reference with a second opcode identifying the parameter of the GUI object;

reading the compact HTML page from at least one memory in the device; and

in response to the first and the second opcodes, executing executable codes instructions in the at least one memory to communicate the parameter with the source and to render the GUI object in response to the parameter.

Claim 46 (withdrawn): The method of claim 45, wherein said communicating the parameter with the source comprises receiving a status of the device from an embedded controller, the embedded controller monitoring and controlling the device.

Claim 47 (withdrawn): The method of claim 45, wherein said communicating the parameter with the source comprises receiving a command from an input device.

Claim 48 (withdrawn): The method of claim 47, further comprising executing the executable codes to send the command to the embedded controller.